33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

Agile Systems Engineering – Eight Core Aspects

Rick Dove, Kerry Lunney, Michael Orosz, Mike Yokell

FuSE Agility Project of the Agile Systems & Systems Engineering Working Group

Mission and Needs

“Systems engineering anticipates and effectively responds to an increasingly dynamic and uncertain environment.”

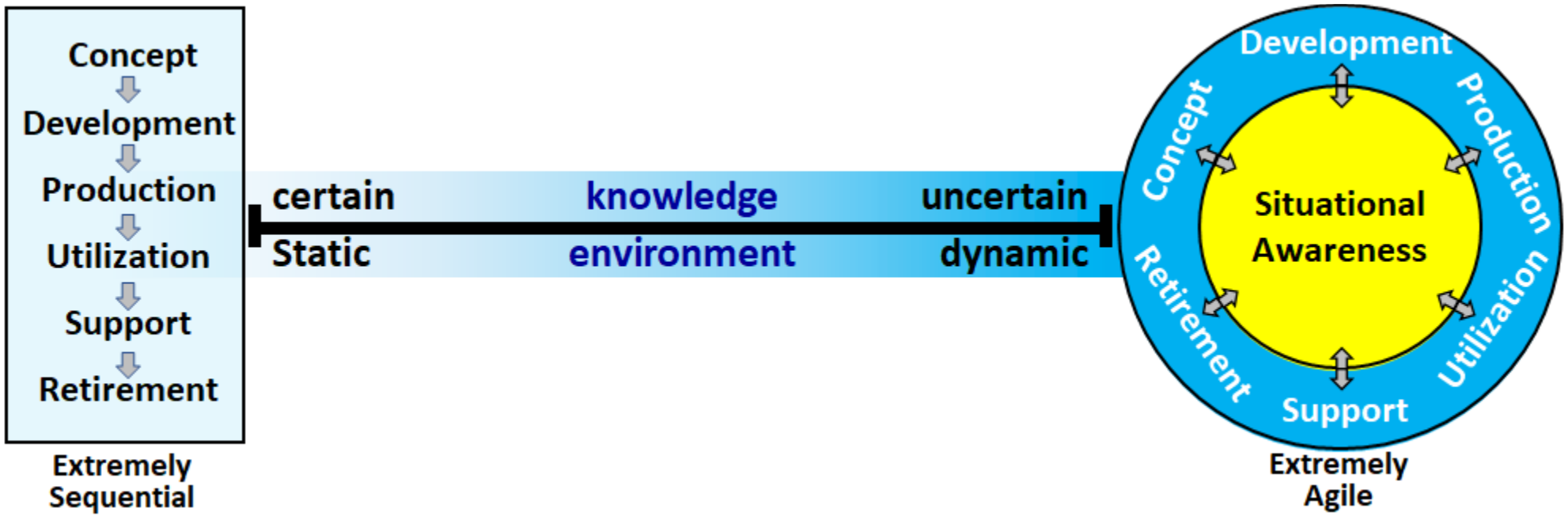
Vision 2035

CAB’s six questions of 2022:

1. **What does it mean to be agile in the context of systems engineering?**
2. **What are the key practices that can make systems engineering agile?**
3. **How can organizations be more agile in their development of systems?**
4. **What benefits can be gained using agile practices for systems engineering?**
5. **What is the relation between agility and model-based systems engineering (MBSE)?**
6. **Are there system characteristics and architectures that make some systems more amenable to agile development and others less so?**

Timely relevance for a work-in-process that began in 1991.

Systems Engineering Life Cycle Spectrum
Sequential to Agile



Agile systems engineering is systems engineering as it is known through ISO/IEC/IEEE standards, the Vee model, and the INCOSE Handbook.

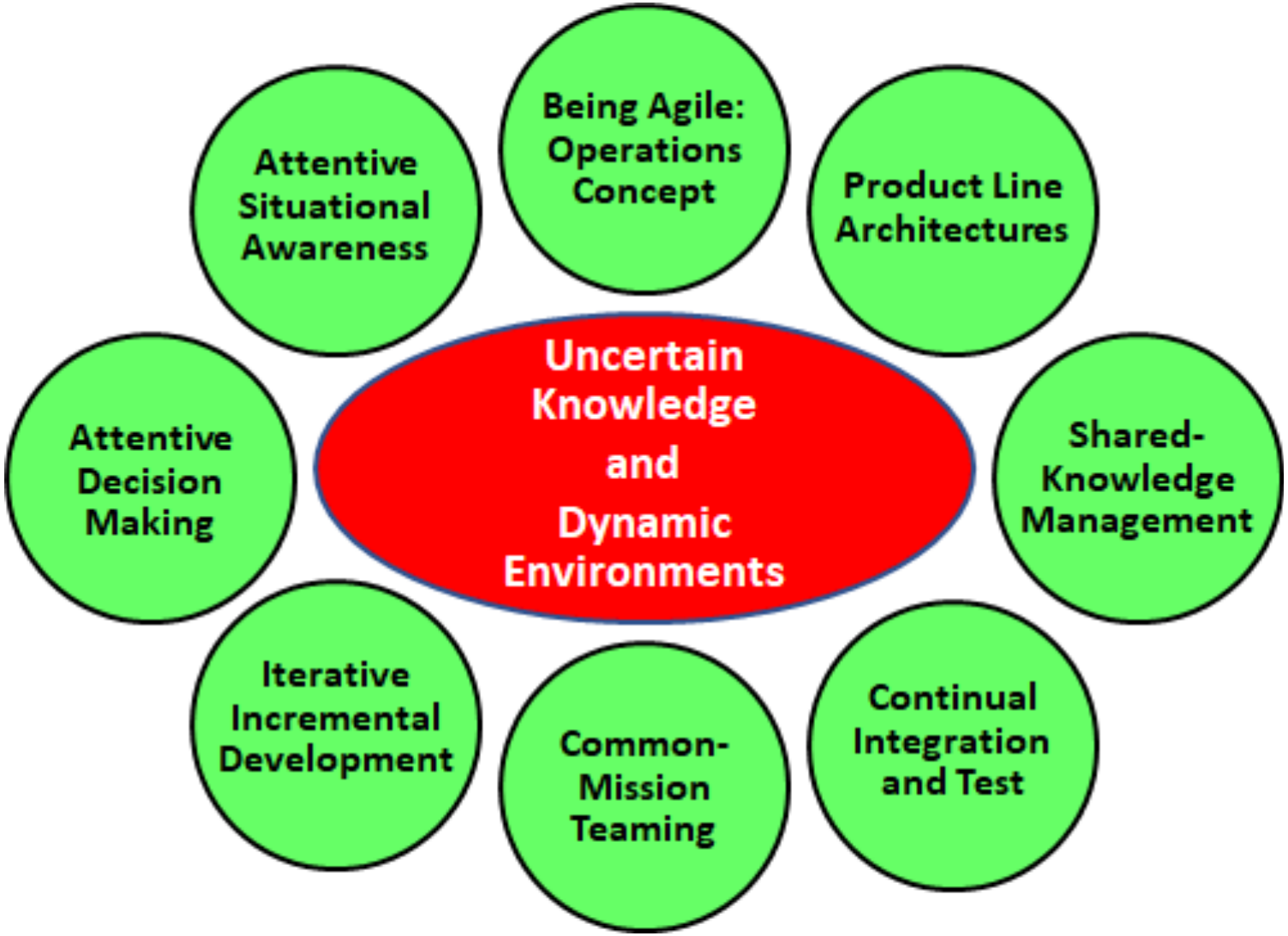
What distinguishes it as "agile" systems engineering is its leverage of situational awareness in driving continual evolution.

Agile systems engineering is a strategy-based method for designing, building, sustaining, and evolving systems when knowledge is uncertain and/or environments are dynamic

Agile System Engineering is a what, not a how.
There are many hows, principally focused on the development phase,
e.g.
Evolutionary Development
Iterative Incremental Development (IID)
Incremental Commitment Spiral Model (ICSM)
et al.

and also many focused on a single engineering domain,
e.g.
Scrum, Kanban, XP, DevOps, et al.

Eight Strategic Aspects That Enable Agility



Product Line Architectures

Needs: Facilitated product and process experimentation, modification, and evolution.

Behaviors: Composable and reconfigurable product and process designs from variations of reusable assets.

Discussion: One fixed process approach won't fit all projects, so an appropriate process should be easy to compose and evolve according to context and usage experience. Variations of reusable assets are built over time as features are modified for different contextual usage.

A hallmark of agile systems engineering is iterative incremental development, which modifies work in process as suitability is repetitively evaluated. The agility of the process depends on the agility of the product – so both process and product can be easily changed.

Iconic Agile Architecture Pattern

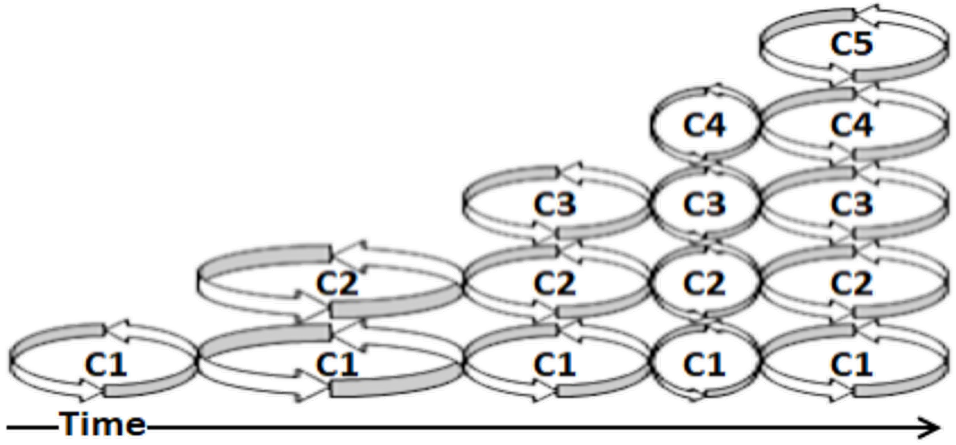
Iterative Incremental Development

Needs: Minimize unexpected rework and maximize quality.

Behaviors: Incremental loops of building, evaluating, correcting, and improving capabilities.

Discussion: Generally increments *create* capabilities and iterations add and augment features to *improve* capabilities.

- Increment cycles are beneficially timed to coordinate events such as integrated testing and evaluation, capability deployment, experimental deployment, or release to production.
- Increments may have constant or variable cadence to accommodate management standards or operational dynamics.
- Iteration cycles are beneficially timed to minimize rework cost as a project learns experimentally and empirically.



Iterative capability improvements (looping) and incremental capability additions (successive columns)

Attentive Situational Awareness

Needs: Timely knowledge of emergent risks and opportunities.

Behaviors: Active monitoring and evaluation of relevant internal and external operational-environment factors.

Discussion: Are you doing things right (internal awareness) and doing the right things (external awareness)? Having the agile capability for timely and cost-effective change does little good if you don't know when that ability should be exercised. Situational awareness can be enhanced with systemic methods and mechanisms.

Alert in-the-moment constant attention

Attentive Decision Making

Needs: Timely corrective and improvement actions.

Behaviors: Systemic linkage of situational awareness to decisive action.

Discussion: Empower decision making at the point of most knowledge. As a counter example, technical debt (a term for knowing something needs correction or improvement but postponing action) is situational awareness without a causal link to prompt action.

John Boyd's OODA loop

Common-Mission Teaming

Needs: Coherent collective pursuit of a common mission.

Behaviors: Engaged collaboration, cooperation, and teaming among all relevant stakeholders.

Discussion: Collaboration, cooperation, and teaming are not synonymous, and need individual support attention. Collaboration is an act of relevant information exchange among individuals, cooperation is an act of optimal give and take among individuals, and teaming is an act of collective endeavor toward a common purpose.

Tightly integrated coherent operation

Shared-Knowledge Management

Needs: Accelerated mutual learning and single source of truth for internal and external stakeholders.

Behaviors: Facilitated communication, collaboration, and knowledge curation.

Discussion: There are two kinds of knowledge to consider. Short time frame operational knowledge: What happened, what's happening, what's planned to happen. Long time frame curated knowledge: what do we know of reusable relevance, e.g., digital artifacts, lessons learned, and proven practices.

Depicted books represent information containers of any kind; but typically digital

Continual Integration & Test

Needs: Early revelation of system integration issues.

Behaviors: Integrated demonstration and test of work-in-process.

Discussion: Discovering integration issues late in development activities can impact cost and schedule with major rework. Synchronizing multiple domain engineering activities via continual integration and test provides faster and clearer insight into potential system integration issues.

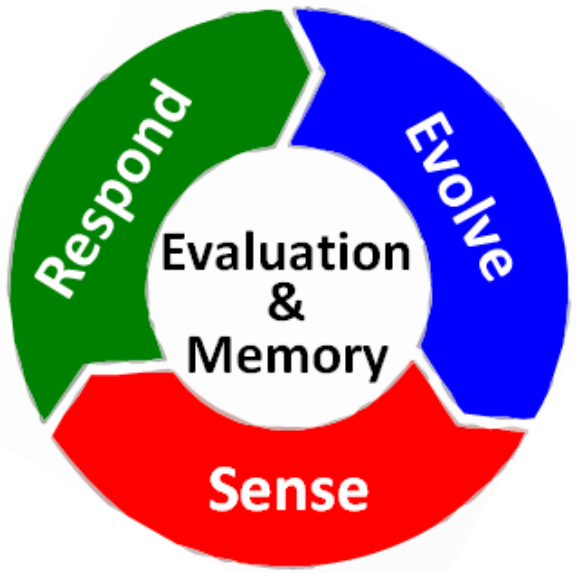
SpaWar iteratively evolving unmanned technology integration platform.

Being Agile: Operations Concept

Needs: Attentive operational response to evolving knowledge and dynamic environments.

Behaviors: Sensing, responding, evolving.

Discussion: Agile systems engineering is not about doing Agile, it is about being agile. Being agile is a behavior, not a procedure – a behavior sensitive to threats and opportunities in the operational environment, decisive when faced with threat or opportunity, and driven to improve these capabilities. Deciding how to implement any of the core aspects, even this one, should be done with sense-respond-evolve principles in mind as aspect objectives.



Three principles that operationalize agility

Product Line Architectures

Needs: Facilitated product and process experimentation, modification, and evolution.

Behaviors: Composable and reconfigurable product and process designs from variations of reusable assets.

Discussion: One fixed process approach won't fit all projects, so an appropriate process should be easy to compose and evolve according to context and usage experience. Variations of reusable assets are built over time as features are modified for different contextual usage.

A hallmark of agile systems engineering is iterative incremental development, which modifies work in process as suitability is repetitively evaluated. The agility of the process depends upon the agility of the product – so both process and product can be easily changed.

Notional Agile Architecture Pattern

Attentive Situational Awareness

Needs: Timely knowledge of emergent risks and opportunities.

Behaviors: Active monitoring and evaluation of relevant internal and external operational-environment factors.

Discussion: Are you doing things right (internal awareness) and doing the right things (external awareness)? Having the agile capability for timely and cost-effective change does little good if you don't know when that ability should be exercised. Situational awareness can be enhanced with systemic methods and mechanisms.

Alert in-the-moment constant attention

Common-Mission Teaming

Needs: Coherent collective pursuit of a common mission.

Behaviors: Engaged collaboration, cooperation, and teaming among all relevant stakeholders.

Discussion: Collaboration, cooperation, and teaming are not synonymous, and need individual support attention. Collaboration is an act of relevant information exchange among individuals, cooperation is an act of optimal give and take among individuals, and teaming is an act of collective endeavor toward a common purpose.

Tightly integrated coherent operation

Continual Integration & Test

Needs: Early revelation of system integration issues.

Behaviors: Integrated demonstration and test of work-in-process.

Discussion: Discovering integration issues late in development activities can impact cost and schedule with major rework. Synchronizing multiple domain engineering activities via continual integration and test provides faster and clearer insight into potential system integration issues.

SpaWar iteratively evolving unmanned technology integration platform.

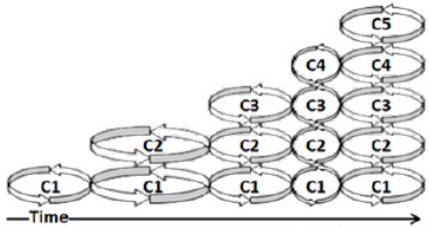
Iterative Incremental Development

Needs: Minimize unexpected rework and maximize quality.

Behaviors: Incremental loops of building, evaluating, correcting, and improving capabilities.

Discussion: Generally increments *create* capabilities and iterations add and augment features to *improve* capabilities.

- Increment cycles are beneficially timed to coordinate events such as integrated testing and evaluation, capability deployment, experimental deployment, or release to production.
- Increments may have constant or variable cadence to accommodate management standards or operational dynamics.
- Iteration cycles are beneficially timed to minimize rework cost as a project learns experimentally and empirically.



Iterative capability improvements (looping) and incremental capability additions (successive development periods)

Attentive Decision Making

Needs: Timely corrective and improvement actions.

Behaviors: Systemic linkage of situational awareness to decisive action.

Discussion: Empower decision making at the point of most knowledge. As a counter example, technical debt (a term for knowing something needs correction or improvement but postponing action) is situational awareness without a causal link to prompt action.

John Boyd's OODA loop

Shared-Knowledge Management

Needs: Accelerated mutual learning and single source of truth for internal and external stakeholders.

Behaviors: Facilitated communication, collaboration, and knowledge curation.

Discussion: There are two kinds of knowledge to consider. Short time frame operational knowledge: what happened, what's happening, what's planned to happen. Long time frame curated knowledge: what do we know of reusable relevance, e.g., digital artifacts, lessons learned, and proven practices.

Depicted books represent information containers of any kind; but typically digital

Being Agile: Operations Concept

Needs: Attentive operational response to evolving knowledge and dynamic environments.

Behaviors: Sensing, responding, evolving.

Discussion: Agile systems engineering is not about doing Agile, it is about being agile. Being agile is a behavior, not a procedure – a behavior sensitive to threats and opportunities in the operational environment, decisive when faced with threat or opportunity, and driven to improve these capabilities. Deciding how to implement any of the core aspects, even this one, should be done with sense-respond-evolve principles in mind as aspect objectives.

Three principles that operationalize agility

Transition and Transformation

Incremental or Big Bang

Each of the aspects can individually improve capability to deal with uncertain knowledge and dynamic environments.

But to have something intended as an agile <any-kind-of> engineering process requires multiple aspects operating in concert.

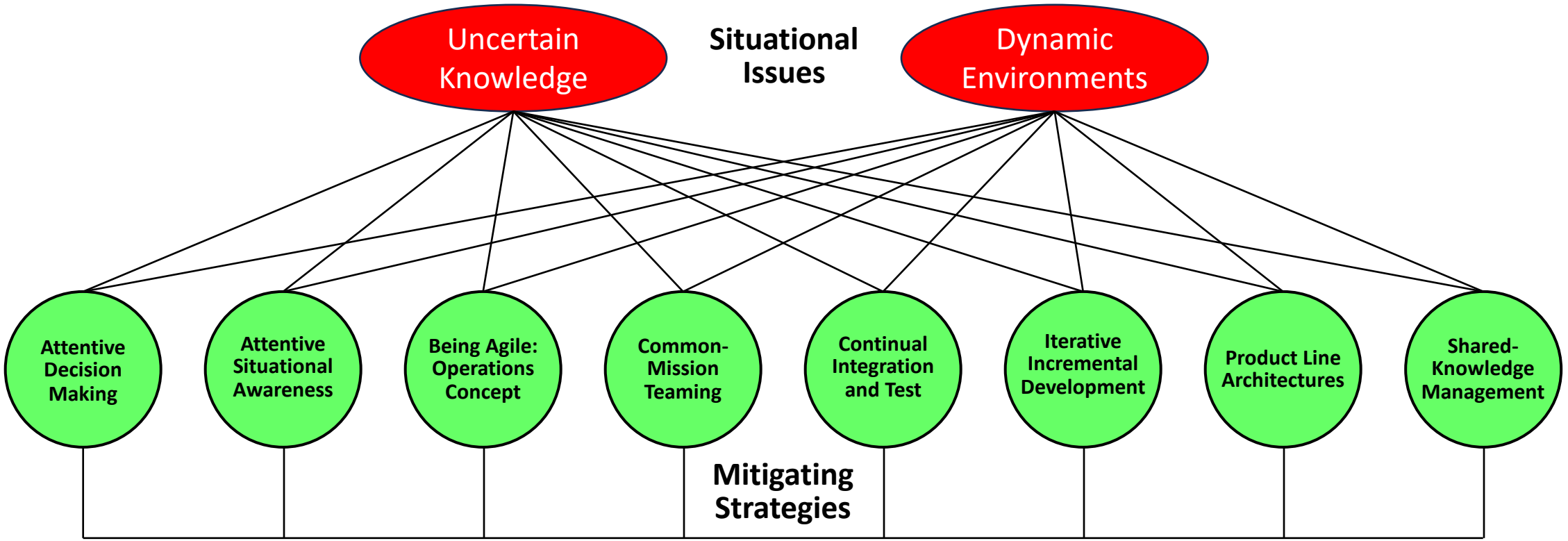
Incremental approach:

work on recognized pain/gain points, the low hanging fruit ready for change .

Big Bang approach:

choose an off-the-shelf framework (e.g., SAFe et al.) and tailor your practice with the aspects.

Synergistic Relationship Potentials

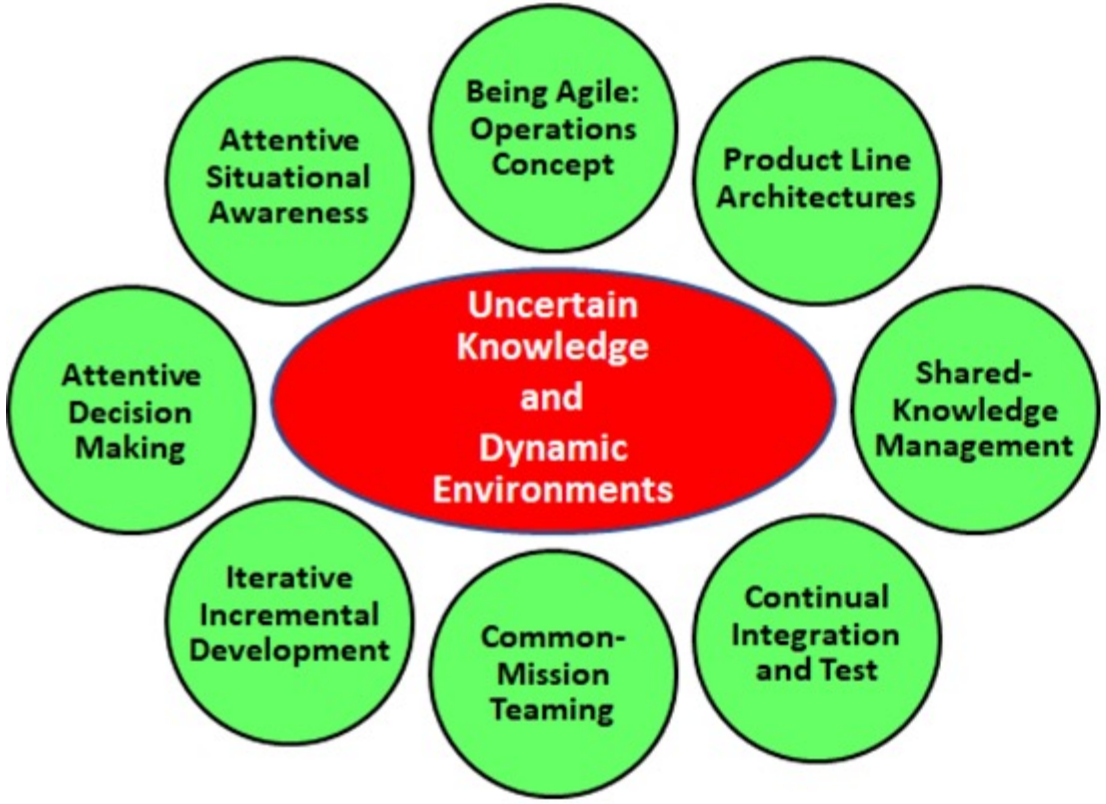


Lines of bidirectional synergy:

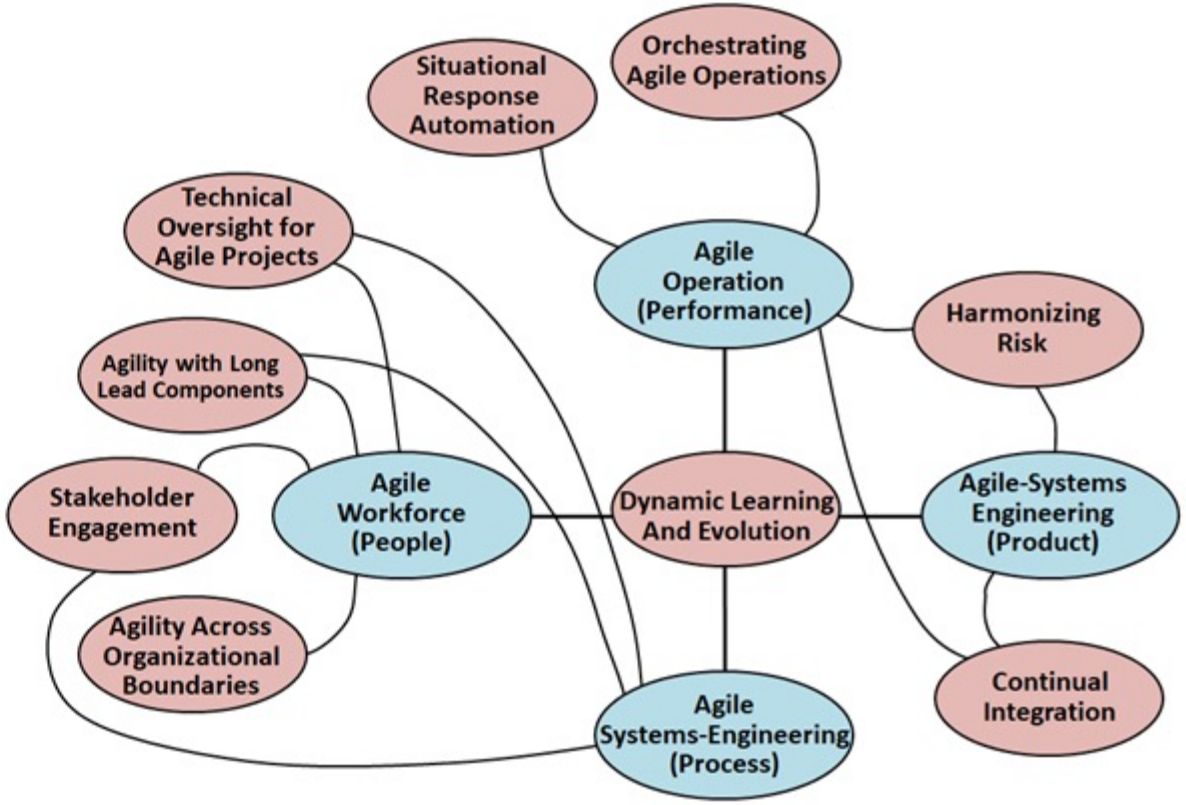
- **Strategies get design purpose from situational issues and provide mitigation impact.**
- **Strategies can help each other for mutual benefit.**

Agile SE is a Journey

Starting & Improving Strategic Aspects



Maturing & Evolving Application Concepts



Large organizations likely have units working in both early and advanced stages